

Japanese Patent Laying-Open No. 05-151179 entitled "Processing  
Assigning System" by Hiroyuki published June 18, 1993.

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-151179  
(43)Date of publication of application : 18.06.1993

(51)Int.Cl. G06F 15/16  
G06F 9/46

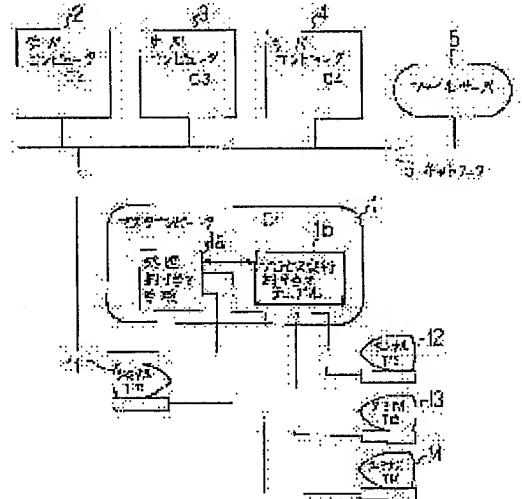
(21)Application number : 03-314500 (71)Applicant : MITSUBISHI ELECTRIC CORP  
(22)Date of filing : 28.11.1991 (72)Inventor : ISHIDA HIROYUKI

## (54) PROCESSING ASSIGNING SYSTEM

### (57)Abstract:

**PURPOSE:** To assign processing uniformly without giving heavy influence such as the remarkable delay of the processing on a processing request from another user even when a large number of processing requests from a specific user are issued.

**CONSTITUTION:** This system is provided with such feature that a master computer 1 which performs the unification of terminals 11-14 is equipped with a processing assigning means 1a, and the means 1a provides upper limit for the number of processes feasible simultaneously at every user by using a process execution assigning table 1b consisting of a process name, an execution assigning computer name, a processing request outgoing user name, and a processing execution standby flag, and makes the execution of the processing request exceeding the upper limit stand by transiently until the number of processes goes less than the upper limit.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-151179

(43)公開日 平成5年(1993)6月18日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 15/16  
9/46

識別記号 庁内整理番号  
3 8 0 Z 9190-5L  
3 6 0 B 8120-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 2(全 12 頁)

(21)出願番号 特願平3-314500

(22)出願日 平成3年(1991)11月28日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 石田 博之

鎌倉市大船五丁目1番1号 三菱電機株式会社情報電子研究所内

(74)代理人 弁理士 高田 守 (外1名)

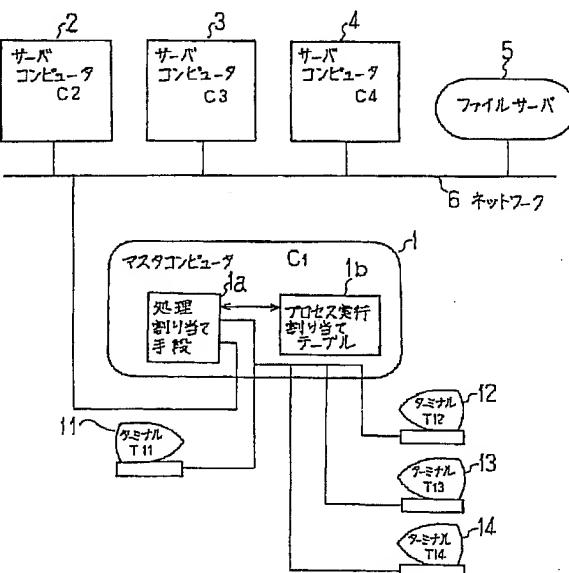
(54)【発明の名称】 处理割り当て方式

(57)【要約】

【構成】 ターミナル11～14の統括を行うマスタコンピュータ1に処理割り当て手段1aを具備し、該手段1aは、プロセス名、実行割り当てコンピュータ名、処理要求発信ユーザ名、処理実行待機フラグからなるプロセス実行割り当てテーブル1bを用い、ユーザごとに同時実行可能プロセス数の上限を設け、上限を越える処理要求に対しては、いずれかのプロセスが終了して上限を下回るまで、実行を一時待機させることを特徴とする。

【効果】 特定ユーザの多量の処理要求があっても、他のユーザからの処理要求に対して処理の大幅な遅れ等の多大な影響を与えることなく処理を均等に割り当てることが可能となる。

分散処理システムの構成例



1

**【特許請求の範囲】****【請求項1】** 以下の要素を有する処理割り当て方式

(a) 処理を実行する複数の処理手段、(b) ユーザからの処理要求を受け付ける入力手段、(c) ひとりのユーザが同時に実行できる処理数を定めた同時実行処理数に基づいて、上記入力手段で受け付けたユーザからの処理要求を上記複数の処理手段のいずれかに割り当てる処理割り当て手段。

**【請求項2】** 上記処理割り当て手段は、少なくとも以下のいずれかに応じて異なる値をとる同時実行処理数を用いることを特徴とする請求項1記載の処理割り当て方式

(a) 複数の処理手段で実行中の処理数、(b) 各ユーザ、(c) 処理要求の優先度。

**【発明の詳細な説明】****【0001】**

**【産業上の利用分野】** 本発明は、分散処理システムにおける処理割り当て方式に関するものである。

**【0002】**

**【従来の技術】** 分散処理システムにおける従来の処理割り当て方式は、処理要求に対して実行可能なコンピュータが複数存在する時に、それら各コンピュータの負荷が均等になり、かつ最も効率的に用いることができるよう、全ての処理要求に対して同時にCPU負荷が最も軽い処理装置に割り当てる方式であった。例としては、図6に示すような、特開昭62-135951号公報に開示されたネットワークシステムにおいて、図7に示すプロセス実行マップを用いて、最も実行処理数の少ない処理装置に割り当てる方式があげられる。

**【0003】** 図6は従来のネットワークシステムの構成例、図7はプロセス実行マップ図、図8はマスタコンピュータ割り込みフローチャート図である。図6において、1、7、8はコンピュータ、5はファイルサーバ、6はネットワーク、11、12、71~73、81~83はターミナルである。また図7において、M1はプロセス実行コンピュータ名、M2はプロセス名、M3はCPU占有タイム、M4はプロセス発信ユーザ名である。

**【0004】** そして、処理割り当ての方式としては、図8に示すように、処理要求を受け付けると、図7に示すプロセス実行テーブルによって、最も実行プロセス数の少ない処理装置を選択し、その処理装置に同時に処理を割り当てる方式であった。

**【0005】**

**【発明が解決しようとする課題】** 従来の処理割り当て方式は、処理要求を受け付けると同時に、最も負荷の軽いコンピュータに割り当てる方式であったため、各コンピュータを効率的に稼働させることができた。しかしユーザ側から見た場合に、あるユーザU11が処理要求を発した時に他のユーザU13が多量の処理要求を発したとすると、ユーザごとのCPU占有時間がU13に偏る状

態となり、結果的にU11からの処理要求の処理終了時間は大幅に遅くなる。

**【0006】** 例えば図6におけるシステムにおいて、図9(a)に示すプロセス実行状態時に、U11からプロセスMA1の処理要求が送られてきた場合、同時にコンピュータST1に送られ実行される。このとき各ユーザの実行プロセス数はほぼ均等であるので、プロセスの処理終了時間の平均は各ユーザのプロセス処理要求数に比例した時間となる。

**【0007】** しかし、図9(b)に示すプロセス実行状態時、すなわち、ユーザU13から多量の処理要求が発せられて、その処理が各コンピュータに均等に割り当てられている状態時に、同様にユーザU11からプロセスMA1の処理要求が送られてきた場合、同時にいずれかのコンピュータに送られ実行されるが、プロセスMA1の処理を受け付けたコンピュータは他に多数のプロセスを実行しているため負荷が非常に重く、結果としてMA1の処理の終了はU11からの処理要求数に比例した処理時間より大幅に遅れる。

**【0008】** つまり、各ユーザから発せられた処理要求を無制限に割り当てるため、特定のユーザから一度に多量の処理要求が発せられた場合に、それらを割り当てられた各コンピュータの負荷が一様に重くなり、他のユーザからの処理要求の処理時間がそのユーザからの処理要求数に比例した時間より大幅に遅れるなどの、多大な影響を被るという問題があった。

**【0009】** 本発明は上記の問題点を解決するためになされたもので、特定のユーザからの多量の処理要求によりユーザごとのCPU占有時間が偏ることなく、各ユーザからの処理要求に対して処理終了時間の大幅な遅れなどの大きな影響のない処理割り当てを行うことを目的とする。

**【0010】**

**【課題を解決するための手段】** 第1の発明に係わる処理割り当て方式では、従来の処理割り当て方式に加え、ユーザごとに同時に実行可能なプロセス数に上限を設定し、ユーザからの処理要求が該ユーザに設定された上記上限を越えた場合、該ユーザからの処理要求を一時待機させる処理割り当て手段を備えることにより、前記問題点を解決する。また、この処理割り当て手段は、各処理要求の処理の終了時には、待機状態にあるプロセスで待機状態を解除し実行させることのできるものがあるかを検索し、その中から最も適当なものを選択し待機状態を解除し実行させる。

**【0011】** 第2の発明に係わる処理割り当て方式では、処理割り当て手段が、以下のいずれかに応じて異なる値の同時実行処理数を用いるものである。

(a) 複数の処理手段で実行中の処理数、(b) 各ユーザ、(c) 処理要求の優先度。

50 **【0012】**

3

【作用】第1の発明における処理割り当て方式は、上記処理割り当て手段により、特定ユーザからの多量の処理要求がCPUを占有することを回避し、ユーザごとのCPU占有率が著しく不均等にならないようにする。

【0013】第2の発明における処理割り当て方式は、上記処理割り当て手段が、同時に実行可能な処理数を、実行中の処理数により動的に変更したり、同時に実行可能な処理数を、ユーザの規模や処理要求の優先度に応じて設定するので、柔軟性のある処理割り当てができる。

【0014】

## 【実施例】

実施例1. 以下、第1の発明の一実施例を図に従って説明する。図1はこの発明を適用する分散処理システムの構成例を示す図である。図中、1はターミナルを統括しサーバコンピュータへの処理割り当て手段1aとプロセス割り当てテーブル1bを有するマスタコンピュータC1、2~4はコンパイル等の処理を行うサーバコンピュータC2、C3、C4、11~14はターミナル、5はファイルサーバ、6はネットワークである。

【0015】図2は、プロセス処理割り当てテーブル1bである。テーブル中のM1はプロセス名、M2は該プロセスの実行を割り当てられたサーバコンピュータ名、M3は該プロセスの処理要求を発したユーザ名、M4は該プロセスが実行中であるか実行待機中であるかを表すフラグである。このフラグを加えたことにより、一人のユーザが投入したプロセス処理要求が同時に実行できるプロセス数の上限に達した際に、該要求を実行待機状態におくことが可能となり、また、実行待機状態にあるプロセスの待機状態を解除し実行させることが可能となる。また、これらのフォーマットにより、あるユーザの実行プロセス数と実行待機プロセス数、プロセスの処理要求発生順序などを導き出すことも可能となる。

【0016】次に上記実施例1の処理の流れを説明する。1ユーザあたりの同時実行プロセスの上限値をPmaxとし、ここではその値を3とする。ユーザU12よりプロセスMULT1の実行指令が入力される。この時のマスタコンピュータ1の動きを図3に沿って説明する。

【0017】まずステップS10でプロセスの実行指令であるかどうかがチェックされる。この場合実行指令であるのでステップS11でプロセス発信ユーザU12の実行プロセス数Pu12をチェックする。次にステップS12では、Pu12がPmaxに達していないかチェックする。ここでは

$Pu12 (=0) < Pmax (=3)$

が成り立つので、ステップS14に分岐しプロセス実行コンピュータの選定を行う。ここでプロセス実行コンピュータとして4が選択され、ステップS15でプロセス実行割り当てテーブルに

[MULT1 : C4 : U12 : T]

が登録される。そして選択されたコンピュータ4にプロセスMULT1の実行が送信される。

【0018】プロセスMULT1が終了するとコンピュータ4はマスタコンピュータ1にプロセスMULT1の実行終了を送信する。マスタコンピュータ1はコンピュータ4からのプロセス終了の送信を受けて、図3のステップS20でプロセス終了の受信を判断して、プロセス実行割り当てテーブルを参照し該プロセスの発信ユーザ名U12を得てそれを一時記憶し、該プロセスの終了処理を行う。そしてプロセス実行割り当てテーブルから該プロセスのデータ

[MULT1 : C4 : U12 : T]

を削除する。次にステップS25でプロセス実行割り当てテーブルをもちいて、一時記憶したユーザ名のU12から発せられた実行待機プロセスの有無を調べる。ここでは、ユーザU12から発せられた実行待機プロセスが存在しないので、処理を終了する。

【0019】次に、実行処理命令を発したターミナルからのプロセスが既にユーザあたりの同時実行プロセス数の上限Pmaxに達している時の、処理割り当ての流れについて説明する。ユーザU13よりプロセスMA1の実行指令が入力される。この時のマスタコンピュータ1の動きを前記同様図3に沿って説明する。

【0020】まずステップS10でプロセスの実行指令であるかどうかがチェックされる。この場合実行指令であるのでステップS11でプロセス発信ユーザU13の実行プロセス数Pu13をチェックする。次にステップS12では、Pu13がPmaxに達していないかチェックする。ここでは

$Pu13 (=3) < Pmax (=3)$

が成り立たないので、ステップS13に分岐しプロセス実行割り当てテーブルに

[MA1 : : U13 : F]

で登録され、処理が終了する。

【0021】最後に実行待機プロセスが待機状態を解かれ実行される際の流れを説明する。コンピュータ3よりプロセスCOMP1の終了が送信されてくる。マスタコンピュータ1はコンピュータ3からのプロセス終了の送信を受けて、図3のステップS20でプロセス終了の受信を判断して、プロセス実行割り当てテーブルを参照し該プロセスの発信ユーザ名U13を得てそれを一時記憶し、該プロセスの終了処理を行う。そしてプロセス実行割り当てテーブルから該プロセスのデータ

[COMP1 : C3 : U13 : T]

を削除する。このときユーザU13から発せられた実行中のプロセス数Pu13は1減少し2( $< Pmax$ )となるので、ユーザU13からのプロセスは実行可能状態となる。そこでステップS25でプロセス実行割り当てテーブルから、一時記憶したユーザU13から発せられた実行待機プロセスの有無を調べる。ここでは、ユーザU

50

13から発せられた実行待機プロセスが存在するので、ステップS26でそれらのうち最も待機時間の長いプロセスを選択する。ここでは、プロセス

【MA1 : : U13 : F】

が選択される。そしてステップS27でプロセスを実行させるコンピュータを選択し、プロセス実行割り当てマップを以下のように修正し、プロセスMA1の実行をコンピュータ2に送信する。

【MA1 : : U13 : F】 → 【MA1 : C2 : U1  
3 : T】

【0022】実施例2. 次に、第2の発明の一実施例について説明する。上記実施例1は、ユーザあたりの同時に実行できる処理要求の上限値Pmaxを固定した処理割り当て方式であるが、コンピュータの負荷状態に応じてPmaxを動的に変更すれば、より柔軟で効率的な処理割り当てを行うことができる。図4は、本実施例による処理割り当てフローチャート中のプロセス実行・待機判定部（図3のS12に相当する）の一例である。この例においては、全コンピュータの各実行プロセス数がSmrn未満であればPmaxは無制限、全コンピュータの各実行プロセス数が、Smrn以上であればPmaxは各Punで凍結（つまり以降全ての処理要求を待機させる）、それ以外であれば定義されたPmaxの値となる。そしてPmaxの動的な変更に対応した処理割り当てフローチャート中の待機プロセス実行部の一例を図5に示す（図3のS22～S29に相当する）。さらにプロセス実行・待機判定の条件を様々に変更すれば、また違った効果を生むことができる。

【0023】実施例3. また、上記実施例1ではPmaxはすべてのユーザに対して同一であるが、この値を各ユーザに対応して異なる値としてもよい。たとえばユーザU11が小口ユーザであり、ユーザU13が大口ユーザであることがあらかじめわかっている場合、ユーザU11用のPmaxは3とし、ユーザU13用のPmaxは10としてもかまわない。

【0024】実施例4. また、ユーザからの処理要求に優先度が付けられるようなシステムにおいては、各ユーザの各優先度ごとにPmaxを定めてもよい。たとえば、優先度に1と2がある場合、優先度1の処理要求に対しては各ユーザのPmaxを5とし、優先度2の処理要求に対しては各ユーザのPmaxを3としてもよい。

【0025】実施例5. また、上記実施例1、2、3、4に記載したPmaxの決定方法を適当に組みあわせてもかまわない。

【0026】以上、上記実施例1～5では、プログラムのコンパイル等の処理を行う複数のサーバコンピュータと、ユーザからの処理入力を受け付ける入力装置と処理結果を表示するディスプレイ装置を持った複数のターミナルと、ターミナルを統括するマスタコンピュータがネットワークによって接続された分散処理システムにおいて

て、マスタコンピュータに処理割り当て手段を具備し、該手段はあらかじめユーザごとに設定された同時に実行可能なプロセス数の上限値に基づいて、特定ユーザからの処理要求が該ユーザに設定された上記上限値を越えた場合、該ユーザからの処理要求を一時待機させることを特徴とする処理割り当て方式を説明した。

【0027】

【発明の効果】以上のように、第1の発明によれば、分散システムにおいて、特定ユーザの多量の処理要求があっても、他のユーザの処理要求に対して処理時間の大幅な遅れ等の多大な影響を与えることなく、処理を均等に割り当てることが可能となる。

【0028】また、第2の発明によれば、同時に実行可能なプロセス数の上限値が、所定の状態によって動的変更されたり、所定の条件により値が異なるようにできるので柔軟な処理割り当てが可能になる。

【図面の簡単な説明】

【図1】本発明を適用する分散処理システムの構成例を示す図である。

【図2】本発明のプロセス実行割り当てテーブル図である。

【図3】本発明の実施例1のプロセス実行割り当てフローチャート図である。

【図4】実施例2におけるプロセス実行割り当てフローチャート中のプロセス実行・待機判定部を示す図である。

【図5】実施例2におけるプロセス実行割り当てフローチャート中の待機プロセス実行部を示す図である。

【図6】従来技術における分散処理システムの構成例を示す図である。

【図7】従来技術におけるプロセス実行マップ図である。

【図8】従来技術における処理の流れ図である。

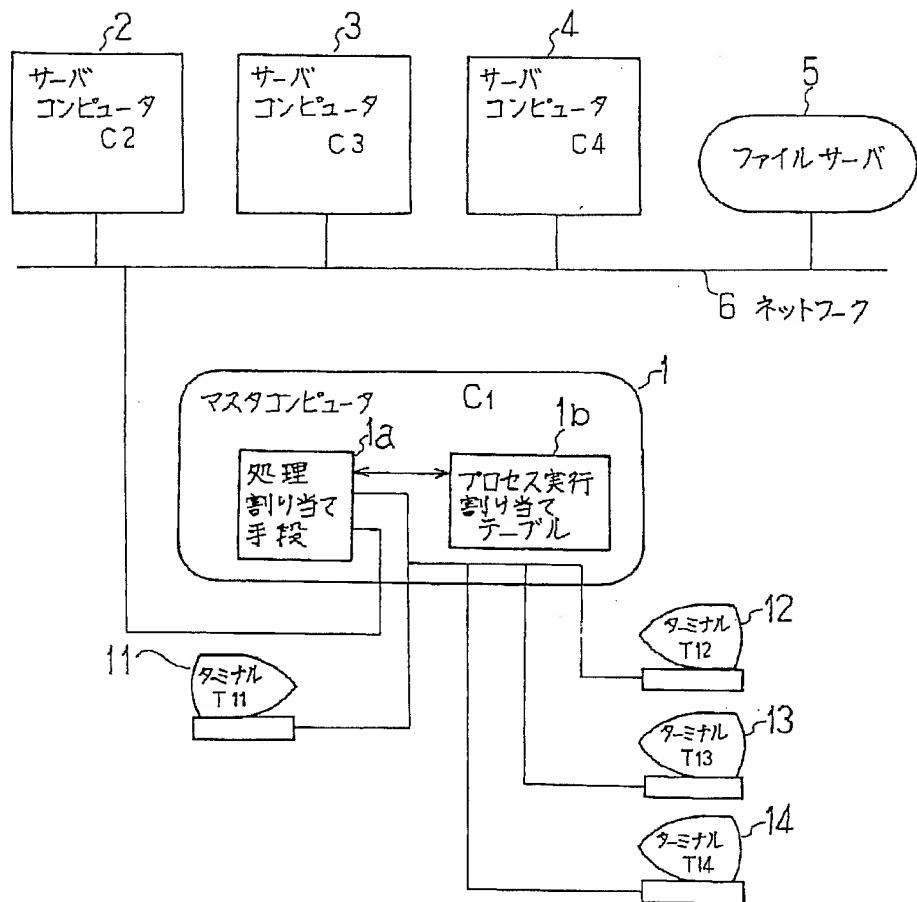
【図9】発明が解決しようとする課題の説明に用いるプロセス実行割り当て状態図である。

【符号の説明】

- 1 マスタコンピュータ
- 1a 処理割り当て手段
- 1b プロセス実行割り当てテーブル
- 2～4 サーバコンピュータ
- 11～14 ターミナル
- 5 ファイルサーバ
- 6 ネットワーク
- 7、8 コンピュータ
- 71～73、81～83 ターミナル
- M1 プロセス名
- M2 実行割り当てコンピュータ
- M3 発信テーブル
- M4 実行フラグ

【図1】

## 分散処理システムの構成例



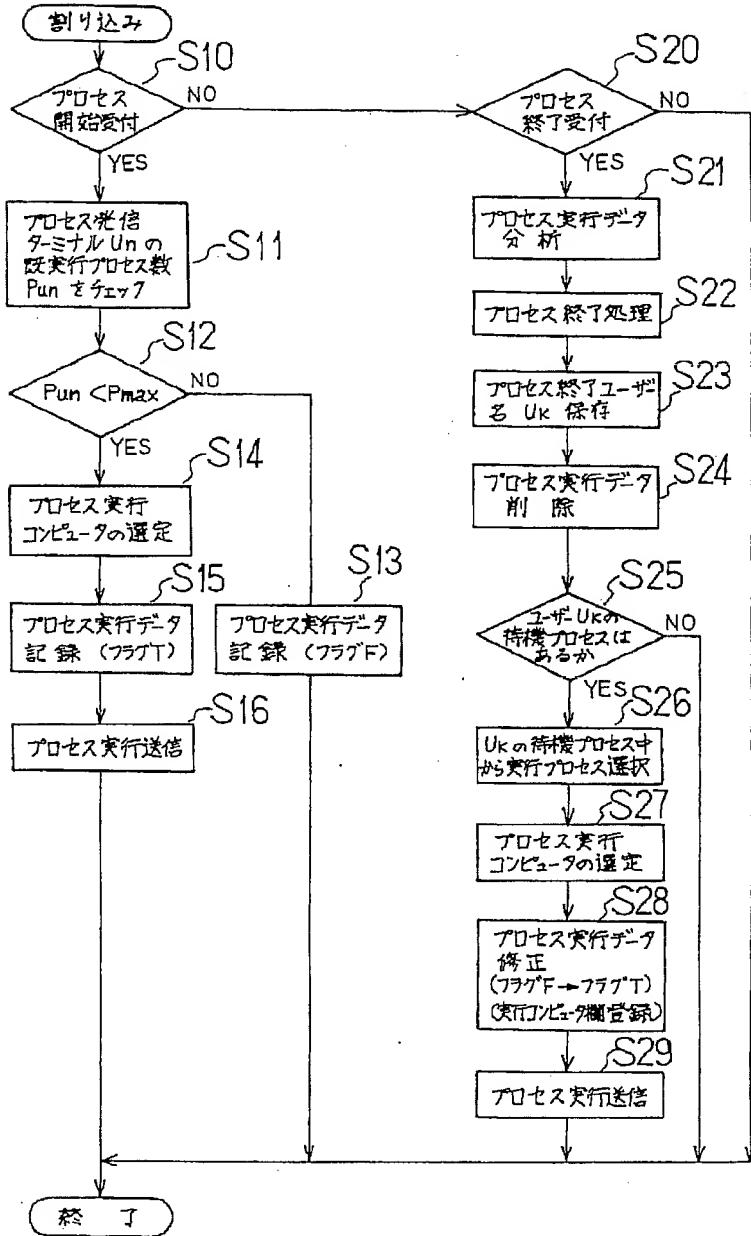
【図2】

プロセス実行割り当てテーブル～1b

プロセス名	実行割り当てコンピュータ	発信ユーザ	実行フラグ
M E 1	C 2	U11	T
COMP1	C 3	U13	T
CALC 1	C 4	U11	T
CALC 2	C 2	U11	T
PLUS1	C 3	U14	T
MULT1	C 4	U12	T
PLUS2	C 2	U13	T
CALC3		U11	F
M X 2	C3	U13	T
M E 2	C4	U14	T
MA 1		U13	F

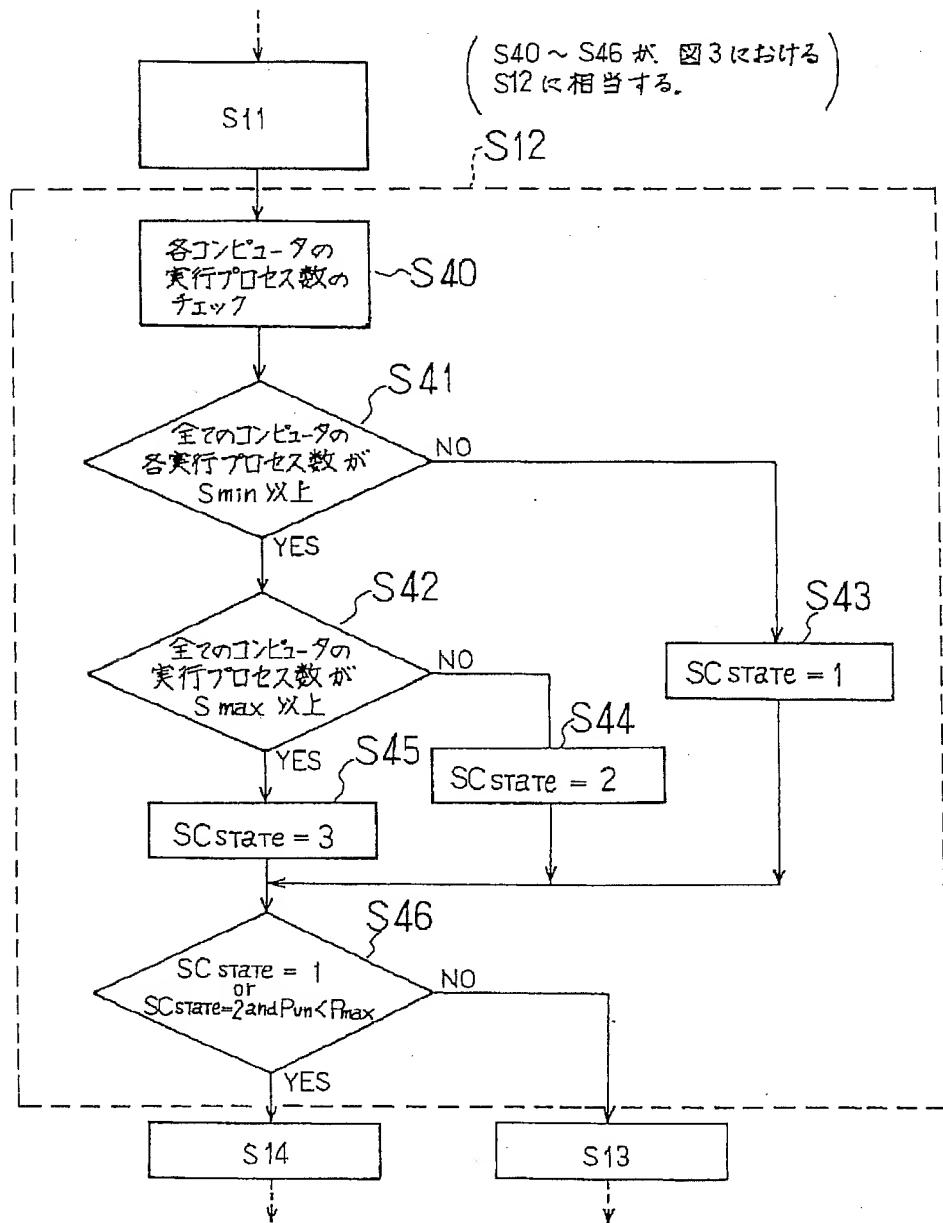
【図3】

## マスター・コンピュータ処理割り当てフローチャート図(実施例1)



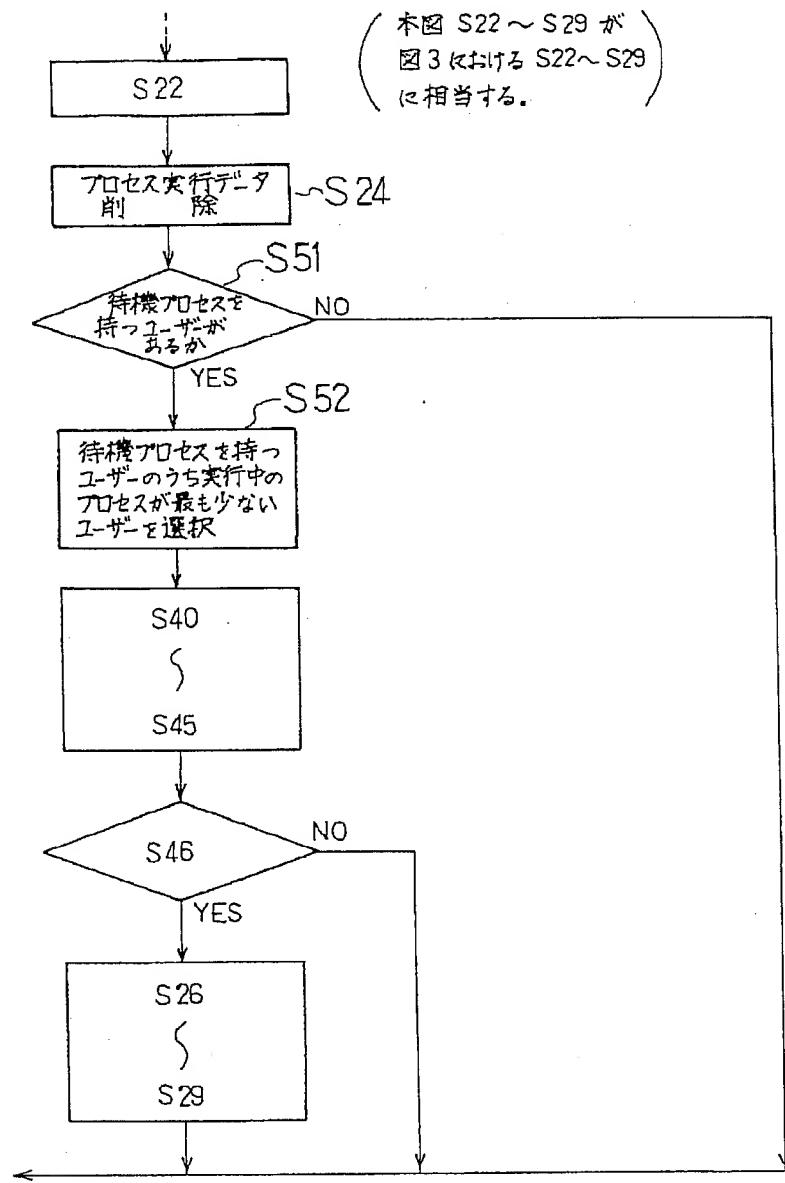
【図4】

実施例2における処理割り当てフローチャート中のプロセス実行・待機判定部



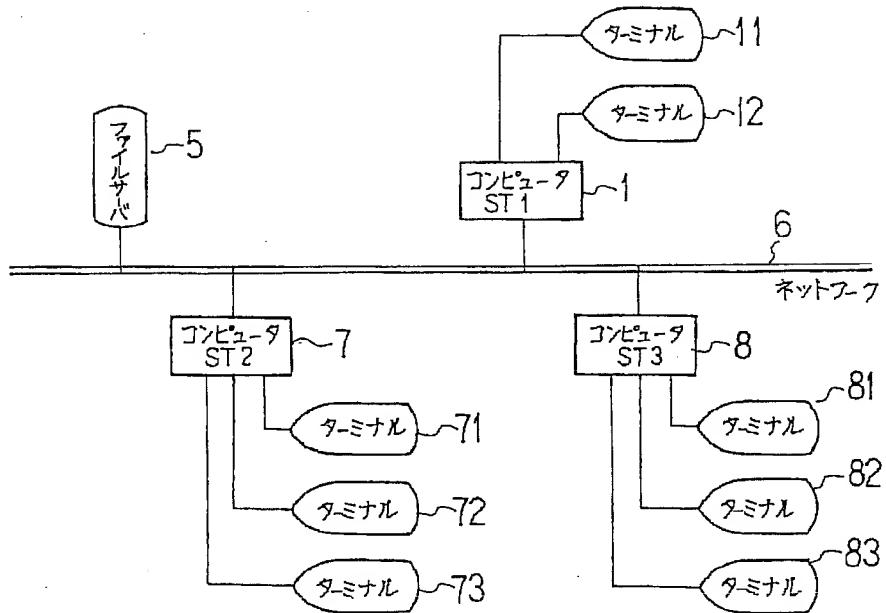
【図5】

実施例2における処理割り当てフローチャート中の待機プロセス実行部



【図6】

## 従来技術におけるネットワークシステムの構成図



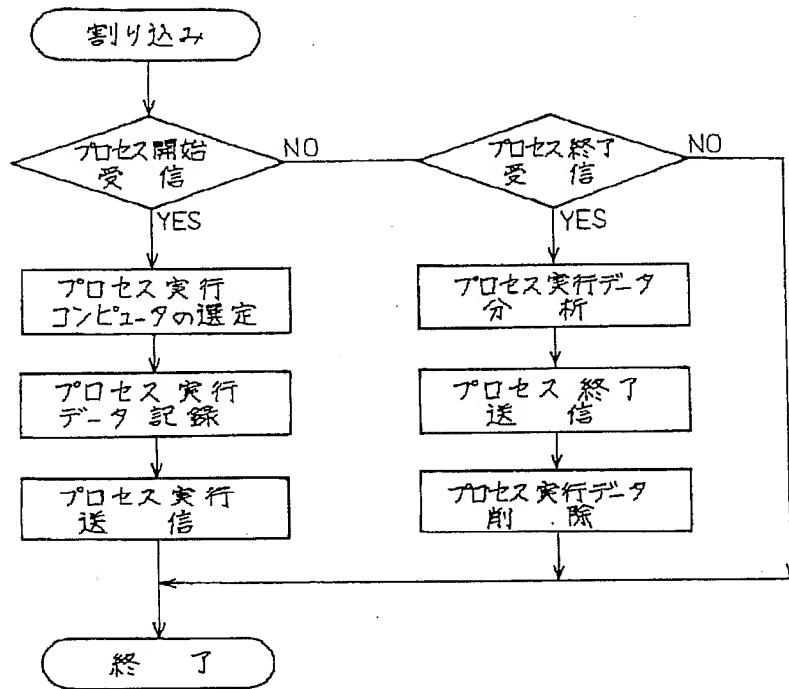
【図7】

## 従来技術におけるプロセス実行マップ

M1	M2	M3	M4
実行コンピュータ	プロセス	リアルタイム	プロセス発信ユーザ
ST 1	CALC 1	0.01	U13
ST 1	FFT 1	0.10	U12
ST 1	CALC 2	1.02	U11
ST 2	FFT 2	0.01	U13
ST 2	C-COMP	2.05	U11
ST 2	FEM 1	60.30	U13
ST 3	FEM 2	0.05	U12
ST 3	FEM 3	0.01	U13

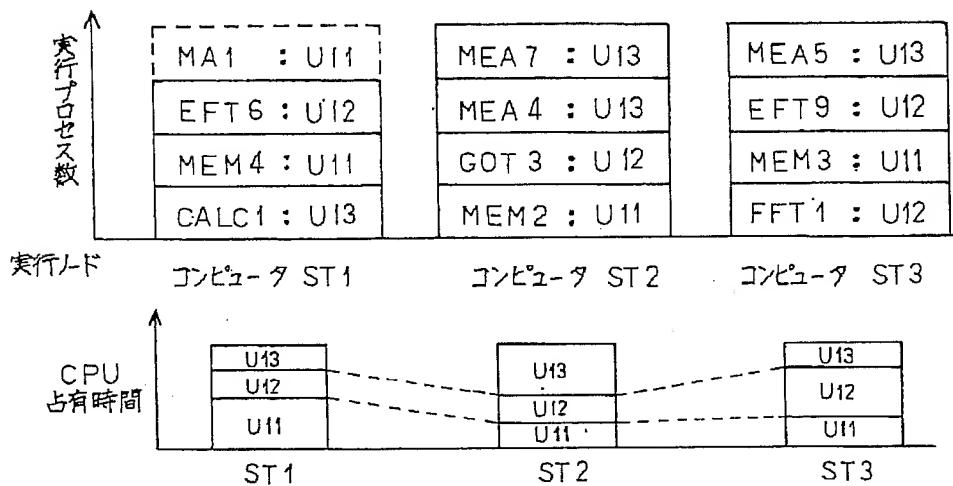
【図8】

従来技術におけるマスター・マスター・コンピュータ割り込みフローチャート図

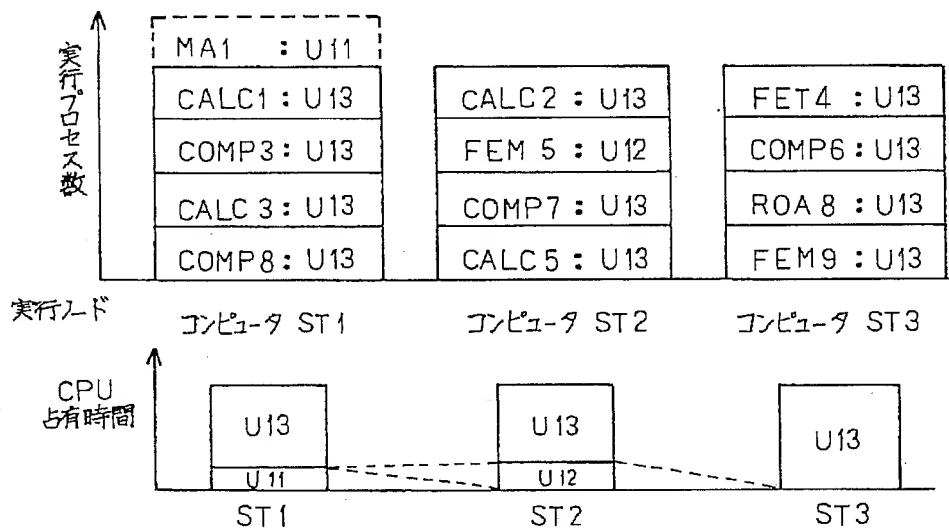


【図9】

(a) 各ユーザごとのCPU占有時間がほぼ均等なプロセス実行状態図



(b) ユーザU13からの多量の処理要求によりCPU占有時間が偏ったプロセス実行状態図



オンライン発送

整理番号: 発送番号: 517931 発送日: 平成20年 9月 2日 1

弓用非特許文献



1035353

特許出願の番号

特願 2002-582366

作成日

平成20年 8月 21日

作成者

上嶋 裕樹 3364 5M00

発明の名称

ファイルアーカイブ